

Improving the Finite-Length Performance of Spatially Coupled LDPC Codes by Connecting Multiple Code Chains

Pablo M. Olmos *Member, IEEE*, David G. M. Mitchell, *Member, IEEE*, Dmitri Truhachev, *Member, IEEE*, Daniel J. Costello, Jr., *Life Fellow, IEEE*

Abstract—In this paper, we analyze the finite-length performance of codes on graphs constructed by connecting spatially coupled low-density parity-check (SC-LDPC) code chains. Successive (peeling) decoding is considered for the binary erasure channel (BEC). The evolution of the undecoded portion of the bipartite graph remaining after each iteration is analyzed as a dynamical system. When connecting short SC-LDPC chains, we show that, in addition to superior iterative decoding thresholds, connected chain ensembles have better finite-length performance than single chain ensembles of the same rate and length. In addition, we present a novel encoding/transmission scheme to improve the performance of a system using long SC-LDPC chains, where, instead of transmitting codewords corresponding to a single SC-LDPC chain independently, we connect consecutive chains in a multi-layer format to form a connected chain ensemble. We refer to such a transmission scheme to as continuous chain (CC) transmission of SC-LDPC codes. We show that CC transmission can be implemented with no significant increase in encoding/decoding complexity or decoding delay with respect a system using a single SC-LDPC code chain for encoding.

Index Terms—codes on graphs, spatially coupled LDPC codes, iterative decoding thresholds, finite-length code performance.

I. INTRODUCTION

Low-density parity-check (LDPC) block codes, invented by Gallager in the 1960's [1] and later rediscovered in the 1990's [2], [3], have attracted a great deal of interest due to their potential for near-capacity performance. However, the iterative decoding techniques generally employed for LDPC decoding are suboptimal compared to optimal maximum a posteriori probability (MAP) decoding, which is prohibitively complex for the typical operational lengths of LDPC codes. As a result, the iterative belief propagation (BP) decoding limits (thresholds) of LDPC block codes are worse than their MAP decoding thresholds [4].

In contrast, it has recently been proven that spatially coupled low-density parity-check (SC-LDPC) codes can be designed to

achieve capacity over general binary-input memoryless output-symmetric (BMS) channels under iterative BP decoding [5]. The reason for this behavior is the spatial graph coupling that defines the structure of an SC-LDPC code: the Tanner graph of a block code with M variable nodes is duplicated L times to produce a sequence of identical graphs; following this, the neighboring copies are then connected to form a chain by redirecting (spreading) certain edges. The resulting graph has a so-called “structured irregularity”, where parity check nodes located at both ends of the chain are connected to a smaller number of variable nodes than those in the middle [6]. As a result, the nodes at the ends of the graph form strong sub-codes and the resulting reliable information generated there during BP decoding propagates through the chain toward the center. It has been shown that, for sufficiently large L , the BP decoding thresholds of (l, r) -regular SC-LDPC codes coincide with the MAP decoding thresholds of the underlying (l, r) -regular LDPC block codes [6] [7], which are known to approach capacity with increasing graph density. As a result, the principle of spatial graph coupling has attracted significant attention and has been successfully applied in many other areas of communications and signal processing [8], [9], [10].

Spatial graph coupling need not be limited to the connection of graphs to form a single chain. In [11], [12], more general ensembles were proposed that are constructed by connecting together several individual SC-LDPC code chains. It was demonstrated that, by optimizing the connection points, the lengths of the connected chains, and their densities, ensembles with improved decoding thresholds can be constructed. A particularly interesting example of such a construction is the loop ensemble [11], [12]. For short chain lengths, *e.g.*, L between 5 and 20, the loop ensemble constructed by connecting two (l, r) -regular SC-LDPC code ensembles has significantly better BP thresholds than single component chains of the same rate and code length. However, for long chain lengths, no constructions have yet been discovered for ensembles created by connecting multiple SC-LDPC chains that are able to outperform a single chain of the same rate and code length.

In contrast to most of the performance analysis of SC-LDPC codes that typically focuses on the asymptotic (in code length) behavior of code ensembles (see, *e.g.*, [5] [7] [6], [13]), in this paper we are interested in the *finite-length* performance of SC-LDPC codes. Analysis of the finite-length performance of LDPC codes in the waterfall region of the bit-error rate (BER) curve is typically performed for the case

This work was partially supported by NSF grant CCF11-61754, by TELUS Corporation Canada, and by Spanish government grants MEC TEC2009-14504-C02-01, TEC2012-38800-C03-01, and Consolider-Ingenio 2010 CSD2008-00010.

P. M. Olmos is with the Signal Theory & Communications Department, Universidad Carlos III de Madrid. E-mail: olmos@tsc.uc3m.es

David G. M. Mitchell and Daniel J. Costello, Jr. are with the Electrical Engineering Department, University of Notre Dame, Indiana, USA. E-mail: {david.mitchell, costello.2}@nd.edu

Dmitri Truhachev is with the ECE Dept., Dalhousie University, Halifax, Canada. E-mail: dmitry@dal.ca

This work was presented in part at the IEEE Information Theory Workshop, Sevilla, Spain, Sept. 2013.

of transmission over the binary erasure channel (BEC), where scaling laws relating the finite-length code performance and the LDPC code parameters can be analytically computed [14], [15]. It has also been observed that such scaling laws are essentially valid across all BMS channels [16]. For the BEC, we consider an equivalent formulation to BP called peeling decoding (PD) [17]. PD iteratively removes variable nodes with known (decoded) values from the Tanner graph. This yields a sequence of graphs whose statistics define both the asymptotic and finite-length properties of the code [18]. An estimate of the PD error probability throughout the decoding process can be obtained based on the expected or mean evolution of the number of degree-one check nodes, as well as the variance around this evolution [14]. Indeed, the PD error probability is dominated by local minima in the expected evolution of the number of degree-one check nodes [19]. We refer to these minima as critical points.

Scaling laws for single SC-LDPC chains have been recently proposed for the BEC in [20], [21], for large L values, and in [22] for small values of L . In this paper, we investigate the finite-length properties of a variety of single and connected SC-LDPC code chains. An important result is that the finite-length performance of short and long SC-LDPC chains is governed by scaling laws with very different characteristics. For short chains, where the code rate is lower, the expected evolution of degree-one check nodes displays only a single critical point, similar to regular LDPC codes [19]; however, for (higher rate) long chains, critical points span the entire expected evolution curve and thus the decoder might fail with roughly uniform probability at any point in the process. From this point of view, shorter chains can be considered to be more robust against decoding failures. This result has not yet been presented in the literature and it is the first main contribution of this paper. Note the important trade-off from the design perspective, already pointed out in [23], [24]: while channel capacity can only be achieved in the limit of large L , for a given finite code length $n = ML$ and a minimum acceptable design rate, the performance of a SC-LDPC code with shorter chain length L and larger M may be better than for a code with longer L and smaller M , even though the threshold of the latter code is closer to capacity.

Next, we build on the results of [11], [12], [22] and investigate the performance of connected SC-LDPC code chains using PD. In particular, we study the finite-length properties of the connected “loop” ensemble. We show that, for short chain lengths, not only the loop has a threshold closer to capacity than the single chain ensemble, also it has better finite-length performance compared to the single chain ensemble of the same rate and code length. The expected evolution of degree-one check nodes is analytically computed for both ensembles and we show that, in each case, the performance is dominated by a single critical point. Besides, we find that there is little or no improvement with this construction for longer chains. To provide a better understanding of the decoding dynamics of connected chains, we revisit the role of those positions at which two or more chains are linked. To this end, we first consider a modified chain ensemble, consisting of two regions with higher degree variable nodes rather than the

typical construction with regions of lower degree check nodes. We show that this method of constructing an SC-LDPC code chain has important weaknesses and the beneficial effect of spatial coupling is lessened. This behavior can then be used to explain the limitations of the loop ensemble when constructed using long SC-LDPC chains and to give guidelines on how to construct code ensembles with improved performance in this regime.

Based on this new understanding of the decoding dynamics of short and long chains, the next contribution of this paper is to present a new transmission technique using SC-LDPC code chains that improves the finite-length error rate performance, in particular when long chains are used. In the case of a single terminated SC-LDPC code chain, the information is encoded into independent codewords corresponding to unconnected consecutive SC-LDPC chains. We propose to link consecutive chains by creating a dependence between them, thus forming a special pattern corresponding to an SC-LDPC code based on connected chains. This coding strategy resembles block Markov superposition encoding [25], first proposed by Cover and El Gamal in [26] for the relay channel. We refer to such an encoding/transmission scheme as continuous chain (CC) transmission. We also propose a way to exploit the two strong sub-codes at both ends of the individual SC-LDPC chains so that the connected chain results in a stronger code of the same rate. The underlying principle is to generate reliable information at various points in the graph - effectively breaking long chains into better protected shorter chains. CC transmission is shown to be feasible for both encoding and decoding. In particular, CC transmission only requires some additional memory in the encoding process and, to effectively implement a windowed decoder [6], [23], a different order in the sequence of transmitted bits compared to transmitting unconnected chains. To the best of our knowledge, this is a new concept in the field of SC-LDPC codes. The final contribution of this paper is to propose a number of designs for SC-LDPC ensembles based on connected chains and optimized for CC transmission. Simulation results for the BEC and the binary input additive white Gaussian noise (BIAWGN) channel show that the finite-length error rate performance improves significantly, even for extremely long code lengths, *e.g.*, $M = 2000$ bits per position and a chain length of $L = 50$.

In Section II we review the construction of a single SC-LDPC code chain and we analyze its finite-length performance. In Section III we introduce a modification to this ensemble, in which strong sub-codes composed of low-degree check nodes are replaced by groups of variable nodes of higher degree. Studying the characteristics of this modified ensemble provides an explanation for the performance improvement obtained when connecting SC-LDPC chains, particularly for the loop construction in Section IV. In Section V we present the CC transmission scheme, propose SC-LDPC code constructions based on connected chains that are well suited for CC transmission, and give simulation results that illustrate its benefits. Finally, we provide some concluding remarks in Section VI, including potential future lines of research on this topic.

II. THE SC-LDPC SINGLE CHAIN ENSEMBLE

In this section we review the construction of a SC-LDPC code chain. We use the random construction of SC-LDPC ensembles presented in [21] and the BEC for most of our analysis. As discussed in the introduction, a finite-length analysis of LDPC codes is feasible for the BEC and, as explained in [21], using the random construction greatly simplifies the analysis. A particularly interesting class of codes can be constructed by means of protographs [27], with important practical advantages [28], [29]; however, these constructions, which can be considered as a subclass of multi-edge type LDPC code [17], significantly complicate the analysis. Nevertheless, the results on connected chains that we derive in the paper can be directly applied to the design of protograph-based SC-LDPC code ensembles.

We start by describing the (l, r) -regular SC-LDPC single chain ensemble. Consider a chain of L uncoupled (l, r) -regular LDPC block codes of length M bits that occupy L consecutive positions. By l (r) we denote the variable (check) node degree of the regular LDPC block code. Each block code has M variable nodes of degree l and $\frac{l}{r}M$ check nodes of degree r . We generate a SC-LDPC code ensemble by spreading $l - 1$ edges per variable node to check nodes associated with nearby codes in the chain.

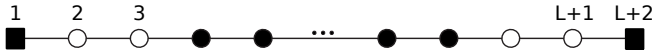


Fig. 1. A single SC-LDPC $\mathcal{C}(3, 6, L)$ chain. White nodes indicate positions where variable nodes are connected to check nodes of degree less than r . Square nodes indicate positions with no variable nodes.

Without loss of generality, we demonstrate the construction of an SC-LDPC code ensemble based on an uncoupled $(3, 6)$ -regular LDPC block code. The $\mathcal{C}(3, 6, L)$ single chain ensemble that we consider is generated as follows. We place $L + 2$ groups of $M/2$ check nodes at positions $u = 1, 2, \dots, L + 2$ and L groups of M variable nodes of degree three at positions $u = 2, \dots, L + 1$. The 3 edges of each variable node located at position u are connected to check nodes at positions $u - 1$, u , and $u + 1$. These check nodes are chosen uniformly at random from the $M/2$ check nodes at each position. At the middle positions $4, \dots, L - 1$, each check node has 6 sockets where edges coming from variable nodes can be connected. So in total, the check nodes in the middle positions have $M/2 \times 6$ sockets and there are $3 \times M$ incoming edges from variable nodes. Thus, after edge spreading all check nodes in the middle positions are of degree 6, since there are no free sockets left.

The $M/2$ check nodes located at positions 1 and $L + 2$ have M incoming edges from variable nodes at positions 2 and $L + 1$, respectively, and thus the number of sockets per check node is only 2, so all check nodes have degree 2 after edge spreading. Similarly, the $M/2$ check nodes at positions 2 and $L + 1$ have $2M$ incoming edges, so the check node degree at these two positions is 4. Overall, the coupled code occupies $L + 2$ positions, labeled from 1 to $L + 2$, but there are variable nodes only in positions 2 to $L + 1$. Any graph generated using this procedure defines a code sample from the $\mathcal{C}(3, 6, L)$ single chain ensemble. Hereafter, we graphically represent a

single chain as shown in Fig. 1, where the round dots indicate positions with variable nodes. White nodes indicate positions where variable nodes are connected to check nodes of degree less than r , i.e., they represent regions with better protection as a result of the lower-degree check nodes. Square nodes indicate positions with no variable nodes.

The $\mathcal{C}(3, 6, L)$ ensemble can be represented in a compact way using an $(L + 2) \times (L + 2)$ binary connectivity matrix \mathbf{T} . $T_{uv} = 1$ means that the M variable nodes at position v are connected with one edge to a randomly chosen check node at position u . Let $d_{c,u}$ denote the sum of elements in the u -th row of \mathbf{T} and $d_{v,u}$ the sum of the elements in the u -th column. The degree of the check nodes at position u is given by $\frac{r}{l}d_{c,u}$, and we get $d_{v,u} = 0$ for the positions with no variables nodes, i.e., the end positions. For example, the 10×10 \mathbf{T} matrix for a single $(3, 6)$ -regular chain with $L = 8$ is

$$\mathbf{T}_{\mathcal{C}(3,6,8)} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}. \quad (1)$$

Throughout the paper, SC-LDPC ensembles will be presented using a $D \times D$ connectivity matrix \mathbf{T} . Any SC-LDPC code based on an (l, r) -regular uncoupled LDPC code with connectivity matrix \mathbf{T} is generated as follows. At position $u = 1, \dots, D$, we place M variable nodes of degree $d_{v,u}$ and $\frac{l}{r}M$ check nodes with $\frac{r}{l}d_{c,u}$ free sockets each. Then we randomly connect the $d_{v,u}$ edges of each variable node at position u to check nodes at positions given by the ones of the u -th column of \mathbf{T} . The design rate of the SC-LDPC ensemble can be computed from the \mathbf{T} matrix. More specifically, the number of check nodes in the Tanner graph is given by $D \frac{l}{r}M$ and the code length is $M \sum_{u=1}^D \mathbb{1}[d_{v,u} > 0]$, where $\mathbb{1}[d_{v,u} > 0]$ equals one if $d_{v,u} > 0$ and zero otherwise. For example, for the $\mathcal{C}(3, 6, L)$ ensemble, $D = L + 2$, $\frac{r}{l} = 2$ and the design rate is

$$r_{\mathcal{C}(3,6,L)} = 1 - \frac{(L + 2) \frac{M}{2}}{M \sum_{u=1}^{L+2} \mathbb{1}[d_{v,u} > 0]} = \frac{1}{2} - \frac{1}{L}. \quad (2)$$

A. Degree Distribution

The concept of code degree distribution (DD) for uncoupled LDPC block codes can be generalized to SC-LDPC codes using the \mathbf{T} matrix. Recall that the right degree of a given edge is defined as the degree of the check it is connected to [18]. Let $R_{j,u}$ be the number of edges with right degree j at position u . Since each variable node has at most one edge connected to a particular position, it is sufficient to know the number V_u of variable nodes per position. Given the connectivity matrix \mathbf{T} , the DD of the SC-LDPC code is, for $u = 1, \dots, D$,

$$V_u = \begin{cases} M & d_{v,u} > 0 \\ 0 & \text{otherwise} \end{cases}, \quad (3)$$

$$R_{j,u} = \begin{cases} d_{c,u}M & j = \frac{r}{l}d_{c,u} \\ 0 & \text{otherwise} \end{cases}. \quad (4)$$

B. Peeling Decoding and expected graph evolution

Now we consider transmission over the BEC and PD at the receiver [17]. We start the PD algorithm by removing from the graph all the variable nodes and edges associated with non-erased symbols, plus any disconnected check nodes. At each iteration, PD looks for a degree-one check node, which is removed along with the variable node it is connected to. In [18], it was shown that if we apply PD to an LDPC code, the sequence of graphs follows a typical path or expected evolution. Based on the graph covariance evolution at those points where the expected evolution of the number of degree-one check nodes presents a local minimum (critical points), *scaling laws* (SLs) for PD were proposed to predict the finite-length performance of LDPC code ensembles in the waterfall region in [14]. To illustrate the main differences in the decoding of short and long SC-LDPC code chains, it suffices to observe the ensemble expected graph evolution over the PD process. The expressions provided are valid for computing the expected graph evolution of any SC-LDPC code constructed according to a given \mathbf{T} matrix.

Assume a BEC with erasure probability ϵ . After initializing PD (iteration $\ell = 0$) by reducing the graph according to the correctly received information, the expected DD at position $u = 1, 2, \dots, D$ is

$$\mathbb{E}[V_u(\ell = 0)] = \epsilon M \mathbb{1}[d_{v,u} > 0], \quad (5)$$

$$\mathbb{E}[R_{j,u}(\ell = 0)] = j \frac{l}{r} M \binom{\frac{r}{l}d_{c,u}}{j} \epsilon^j (1 - \epsilon)^{(\frac{r}{l}d_{c,u} - j)}, \quad (6)$$

for $j \leq \frac{r}{l}d_{c,u}$, where (6) comes from the fact that each one of the $\frac{r}{l}d_{c,u}$ edges of any check node at position u is independently removed with probability ϵ . Let $V_u(\ell)$ and $R_{j,u}(\ell)$ be the (random) DDs at position u after ℓ iterations. Then the normalized (random) DDs at normalized time τ can be described by

$$\tau \doteq \frac{\ell}{\epsilon M}, \quad r_{j,u}(\tau) \doteq \frac{R_{j,u}(\ell)}{\epsilon M}, \quad v_u(\tau) \doteq \frac{V_u(\ell)}{\epsilon M}, \quad (7)$$

for $u = 1, 2, \dots, D$ and $j = 1, 2, \dots, r$. Note that, on average, the total number of PD iterations is $\epsilon M \sum_u \mathbb{1}[d_{v,u} > 0]$, and thus $\tau \in [0, \sum_u \mathbb{1}[d_{v,u} > 0]]$. For the $\mathcal{C}(l, r, L)$ ensemble, $\tau \in [0, L]$.

Let $\hat{r}_{j,u}(\tau)$ and $\hat{v}_u(\tau)$ denote the mean value of the random processes $r_{j,u}(\tau)$ and $v_u(\tau)$ in (7). As shown in [18], $\hat{r}_{j,u}(\tau)$ and $\hat{v}_u(\tau)$ can be computed as the solution to the following system of differential equations:

$$\frac{\partial \hat{r}_{j,u}(\tau)}{\partial \tau} = \mathbb{E}[R_{j,u}(\ell + 1) - R_{j,u}(\ell) | \hat{r}_{q,m}(\ell), \hat{v}_m(\ell) \forall m, q] \quad (8)$$

$$\frac{\partial \hat{v}_u(\tau)}{\partial \tau} = \mathbb{E}[V_u(\ell + 1) - V_u(\ell) | \hat{r}_{q,m}(\ell), \hat{v}_m(\ell) \forall m, q] \quad (9)$$

for $u = 1, 2, \dots, L + 2$ and $j = 1, 2, \dots, r$. Note that the right hand-side of (8) represents the expected change in the number of check nodes of degree j at position u after iteration $\ell + 1$ if the DD at iteration ℓ is exactly at its mean. Further,

the solution of (8) and (9) is unique and, with probability $1 - \mathcal{O}(e^{-\sqrt{M}})$, any particular realization of the normalized DD in (7) deviates from its mean by a factor of less than $M^{-1/6}$ for the initial conditions $\hat{r}_{j,u}(0) = \mathbb{E}[R_{j,u}(\ell = 0)]/\epsilon M$ and $\hat{v}_u(0) = \mathbb{E}[V_u(\ell = 0)]/\epsilon M$. The computation of the expectations in (8) and (9) can be found in Appendix A. Consequently, the ensemble threshold under BP decoding can be computed as the maximum ϵ for which the mean of the normalized number of degree-one check nodes in the graph, given by

$$\hat{r}_1(\tau) \doteq \sum_{m=1}^D \hat{r}_{1,m}(\tau), \quad (10)$$

is positive for any $\tau \in [0, \sum_{i=1}^D \mathbb{1}[d_{v,i} > 0]]$. Based on (10), we can say that $\hat{r}_1(\tau)$ is the mean of the random process $r_1(\tau) = \sum_{m=1}^D r_{1,m}(\tau)$.

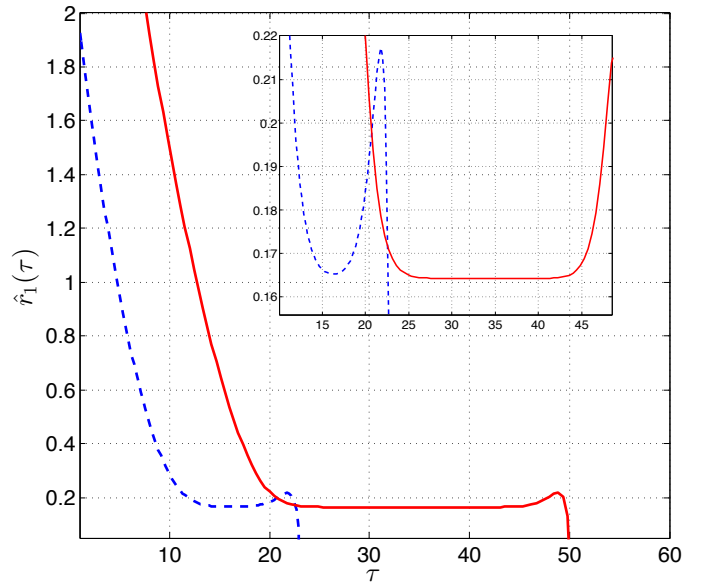


Fig. 2. Solution to $\hat{r}_1(\tau)$ for the ensembles $\mathcal{C}(3, 6, 25)$ (dashed line) and $\mathcal{C}(3, 6, 50)$ (solid line) for $\epsilon = 0.45$. A magnification is shown in the upper-right corner of the figure.

C. Scaling behavior

In Fig. 2, we plot $\hat{r}_1(\tau)$ for the ensembles $\mathcal{C}(3, 6, 25)$ (dashed line) and $\mathcal{C}(3, 6, 50)$ (solid line) and $\epsilon = 0.45$. After magnification, we observe that $\hat{r}_1(\tau)$ for the $\mathcal{C}(3, 6, 25)$ ensemble presents a unique local minima at $\tau^* \approx 16$. As shown in [22], the ensemble average block error probability for short SC-LDPC code chains with a single critical point at a given time τ^* is well predicted by the cumulative probability distribution of the process $r_1(\tau)$ at that time. For sufficiently large M , $r_1(\tau)$ is approximately Gaussian distributed [14] and thus the block error probability can be estimated as follows:

$$P_{\mathcal{C}(l,r,L)} \approx \mathcal{Q}\left(\frac{\sqrt{M}\Delta_\epsilon}{\alpha}\right), \quad (11)$$

where $\Delta_\epsilon = (\epsilon_{\mathcal{C}(3,6,25)} - \epsilon)$, $\epsilon_{\mathcal{C}(3,6,25)}$ is the $\mathcal{C}(3,6,25)$ ensemble's BP threshold and $\alpha = \alpha(l,r)$ is the ratio of the standard deviation of the process $r_1(\tau)$ to its mean $\hat{r}_1(\tau)$, computed at the critical point τ^* .

In contrast, as demonstrated for the $\mathcal{C}(3,6,50)$ ensemble in Fig. 2, for longer chains $r_1(\tau)$ does not display a single critical point, but rather a critical phase in which it remains constant. During this phase, the decoder might fail roughly at any point with uniform probability. As explained in [20], the dynamics of the decoding process are governed by so-called wave decoding [7], [24]. After positions in the middle of the chain have run out of degree-one check nodes, which happens at $\tau \approx 25$ in Fig. 2, recovered information from the boundary positions propagates at constant speed through the chain to the middle positions. These dynamics explain the constant value of $\hat{r}_1(\tau)$ during the so-called steady-state phase. In [20], [21], the block error probability in the steady-state phase is estimated as follows:

$$P_{\mathcal{C}(l,r,L)} \approx 1 - \exp\left(-\frac{\epsilon L \bar{y}}{\mu_0(M, \epsilon, l, r)}\right), \quad (12)$$

where $\bar{y} = \bar{y}(l, r, \epsilon)$ is a quantity that measures how long the decoder remains in the critical phase and μ_0 is the average survival time of the $r_1(\tau)$ process during the critical phase before crossing the zero level, which implies a decoding error. As shown in [21], \bar{y} can be bounded using BP density evolution for the (l, r) -regular LDPC code ensemble. Also, μ_0 can be expressed as follows:

$$\mu_0(l, r, M, \epsilon) \approx \frac{\sqrt{2\pi}}{\theta} \int_0^{\frac{\sqrt{M}\Delta_\epsilon}{\alpha}} \Phi(z) e^{\frac{1}{2}z^2} dz, \quad (13)$$

where $\Phi(z)$ is the c.d.f. of the standard Gaussian distribution, $\mathcal{N}(0, 1)$, and $\theta = \theta(l, r)$ a parameter related to the exponential decay of the covariance of $r_1(\tau)$ with time, i.e., $\text{CoV}[r_1(\tau), r_1(\zeta)] \propto \exp(-\theta|\zeta - \tau|)$ (see [21] for further details).

The scaling law for short SC-LDPC chains in (11) can be considered an extreme case of (12), where wave-like decoding is reduced to a single time instant. For the same distance Δ_ϵ to their respective thresholds and the same number M of bits per position, the process $r_1(\tau)$ has similar statistical properties at the critical point/phase. For instance, in Fig. 2 we observe that the mean height at the respective minima is just slightly higher for the $\mathcal{C}(3, 6, 25)$ case, and by simulation we can confirm that, at these points, the standard deviation of $r_1(\tau)$ is similar in both cases. However, the dynamics of decoding long chains require the process $r_1(\tau)$ to survive a period of time that grows linearly with L and during which its mean is very low. Thus the decoding process is more susceptible to errors.

In Fig. 3, we plot the simulated block error rate (solid lines) of the ensembles $\epsilon_{\mathcal{C}(3,6,50)}$ and $\epsilon_{\mathcal{C}(3,6,25)}$ with $M = 1000$ bits per position. Despite the fact that the ensembles have different rates, the performance comparison is fair, since it is based on the distance to their respective thresholds. We also include the block error rate estimate obtained using (11) and (12) with the following values: $\alpha_{\mathcal{C}(3,6,25)} \approx 0.22$, $\alpha_{\mathcal{C}(3,6,50)} \approx 0.17$,

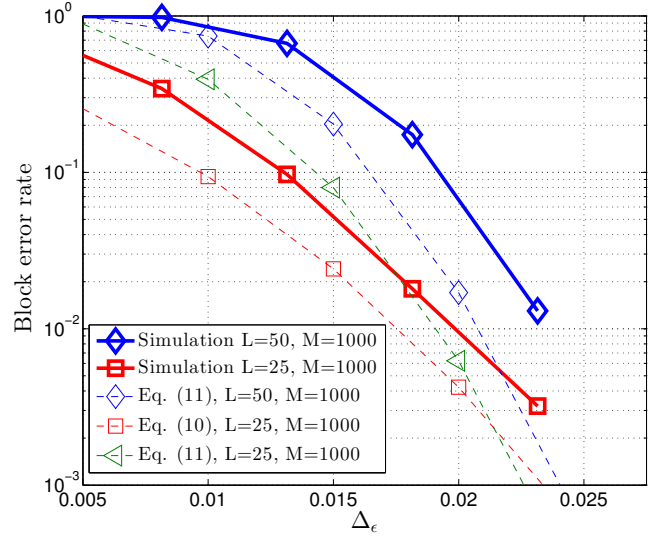


Fig. 3. Simulated block error rate (solid lines) of the ensembles $\epsilon_{\mathcal{C}(3,6,50)}$ and $\epsilon_{\mathcal{C}(3,6,25)}$ with $M = 1000$ bits per position as a function of Δ_ϵ . Also included is the block error rate estimate obtained using (11) and (12) (dashed lines).

and $\theta = 0.59^1$. In (12), \bar{y} can be computed as shown in [20]. We have also included the block error rate estimate for $L = 25$ using the long-chain model in (12), which demonstrates the accuracy of (11) and (12) when used for SC-LDPC code ensembles in the correct regime.

Two important remarks can be made based on Fig. 3. First, at the same distance to their respective thresholds, the ensemble $\mathcal{C}(3, 6, 25)$ provides a significantly lower block error rate. This fact can be observed in both the simulation curves and the estimated block error rate curves. Second, this gain is not just explained by the smaller L value in (12). There is a change in the dynamics of the decoding process and thus qualitative improvement in the robustness of the code. This can be observed in Fig. 3 by comparing the estimated performance for $L = 25$ using (11) (\square marker) and (12) (\diamond marker). This result is of significant practical importance and has not yet been discussed in the literature. In Section V, we exploit this result to improve the finite-length performance of systems based on long SC-LDPC code chains.

III. A MODIFIED SC-LDPC SINGLE CHAIN ENSEMBLE

In this section, we introduce a modification of the chain ensemble $\mathcal{C}(3, 6, L)$ presented above to illustrate the effect of introducing a structured irregularity into the SC-LDPC code chain in a different way. The analysis of this modified code chain allows us to understand, analyze, and design SC-LDPC code ensembles based on connected chains with improved performance, as we demonstrate in Sections IV and V.

¹The difference between the simulated performance and that estimated using scaling laws is mainly because, when using a scaling law such as (11) or (12), we neglect some second order effects that are important for low values of M [14], [20], [22]. For example, in (11) we ignore the fact that, with non-zero probability, $r_1(\tau)$ can equal zero for some $\tau \neq \tau^*$.

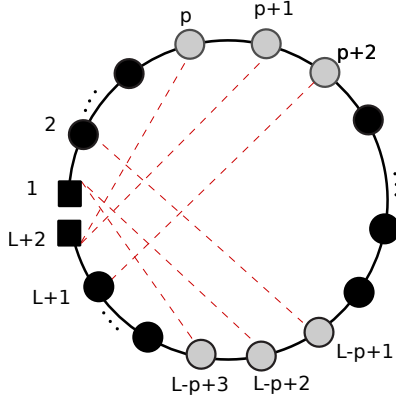


Fig. 4. Graphical representation of the $\tilde{\mathcal{C}}(3, 6, L, p)$ ensemble. Dashed lines represent the additional edges added with respect to the $\mathcal{C}(3, 6, L)$ ensemble and gray dots indicate positions with variable nodes of degree four.

We again consider the $\mathcal{C}(3, 6, L)$ ensemble as an illustrative example. Let $p \in \{2, \dots, \lceil L/2 \rceil\}$ be a position in the code chain that contains variable nodes. We define the ensemble $\tilde{\mathcal{C}}(3, 6, L, p)$ as follows: given the $\mathcal{C}(3, 6, L)$ ensemble, we add one additional edge from each variable node at positions p and $p+1$ to a random and uniformly chosen check node at position $L+2$ and one additional edge between each variable node at position $p+2$ to a check node at position $L+1$. We proceed symmetrically by adding edges in a similar fashion, connecting variable nodes at positions $L-p+3, L-p+2$, and $L-p+1$ to check nodes at the left end of the chain, *i.e.*, positions 1 and 2. These connections are chosen so that we do not increase the check node degree beyond 6. For example, the connectivity matrix \mathbf{T} of the $\tilde{\mathcal{C}}(3, 6, 8, p)$ ensemble for $p = 3$ is

$$\mathbf{T}_{\tilde{\mathcal{C}}(3,6,8,3)} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & \mathbf{1} & \mathbf{1} & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & \mathbf{1} & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & \mathbf{1} & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & \mathbf{1} & \mathbf{1} & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix},$$

where $d_{c,u} = 3$ for $u = 1, 2, \dots, D = L+2$, and thus all check nodes in the graph have degree six. Hence, the graph is *check regular*, as opposed to the standard *variable regular* SC-LDPC code construction.

By including these additional edges, we have *closed* the chain, losing the low-degree check nodes at the ends of the $\mathcal{C}(3, 6, L)$ graphs; however, we have created two regions with better protected symbols, where the variable nodes at three consecutive positions in the chain have degree four rather than three. In Fig. 4, we give a graphical representation of the $\tilde{\mathcal{C}}(3, 6, L, p)$ ensemble. Dashed lines represent the additional edges added with respect to the $\mathcal{C}(3, 6, L)$ ensemble and gray dots indicate positions with variable nodes of degree four.

The expected evolution of the normalized number of degree-

one check nodes in the graph, *i.e.*, $\hat{r}_1(\tau)$ in (10), for the ensemble $\tilde{\mathcal{C}}(3, 6, 50, p)$ with $\epsilon = 0.47$ is shown in Fig. 5. We consider different values for p : $p = 9$, $p = 16$, $p = 22$, and $p = 25$. First, we note that the new BP threshold $\epsilon_{\tilde{\mathcal{C}}(3,6,50,p)}$ is not independent of p , and it is even below 0.47 for $p = 9$ and $p = 16$. For small values of p , the two better protected regions defined by variable nodes of degree four (shown in Fig. 4 by gray circles) are far away from each other. When isolated in this way, unlike the regions protected by the low degree check nodes in the standard $\mathcal{C}(3, 6, 50)$ ensemble, they are too weak to initiate wave-like decoding towards the interior positions of the code. As a consequence, the BP threshold is significantly decreased compared to the threshold of the standard ensemble ($\epsilon_{\mathcal{C}(3,6,L)} \approx 0.488$). In Table I, we present the BP thresholds of the $\tilde{\mathcal{C}}(3, 6, 50, p)$ code ensembles as a function of p . As we increase p , moving the two better protected regions closer together, the threshold quickly approaches 0.488. The maximum threshold is achieved for $p = 25$, which corresponds to a single region of the chain where there are four consecutive positions (positions 25, 26, 27, and 28) with variable node degree larger than 3 (4, 5, 5, and 4, respectively). Note that, in this case, there is no threshold degradation with respect to the $\mathcal{C}(3, 6, 50)$ ensemble. From the finite-length scaling perspective (see Fig. 5), note the presence of a critical point at $\tau \approx 14$ for the ensembles with $p = 22$ and $p = 25$. For the case $p = 22$, this critical point dominates the block error rate; however, for $p = 25$, most errors in the finite-length regime occur in the critical phase corresponding to wave-like decoding towards the interior positions of the code. In Section V-C, we show by simulation that the finite-length performance of the ensembles $\mathcal{C}(3, 6, 50)$ and $\tilde{\mathcal{C}}(3, 6, 50, 25)$ is roughly the same. This is expected because both ensembles have the same threshold and approximately the same scaling behavior, given by (12).

In the following section, we revisit the loop ensemble presented in [11], [12], from the finite-length perspective. The analysis of the $\tilde{\mathcal{C}}(3, 6, 50, p)$ ensemble presented in this section will help us to understand the decoding behavior of the loop ensemble in the large L regime and to propose improved constructions in Section V.

IV. THE LOOP CONNECTED CHAIN ENSEMBLE

We now turn our attention to a more general connected chain SC-LDPC code ensemble, the loop ensemble [11], [12]. We denote it by $\mathcal{L}(l, r, L)$. As before, we use the $(3, 6)$ -regular LDPC code ensemble to demonstrate behavior; however the

p	$\epsilon_{\tilde{\mathcal{C}}(3,6,50,p)}$
9	0.467
16	0.468
22	0.472
24	0.478
25	0.488

TABLE I
BP THRESHOLDS OF THE $\tilde{\mathcal{C}}(3, 6, 50, p)$ CODE ENSEMBLES AS A FUNCTION OF p .

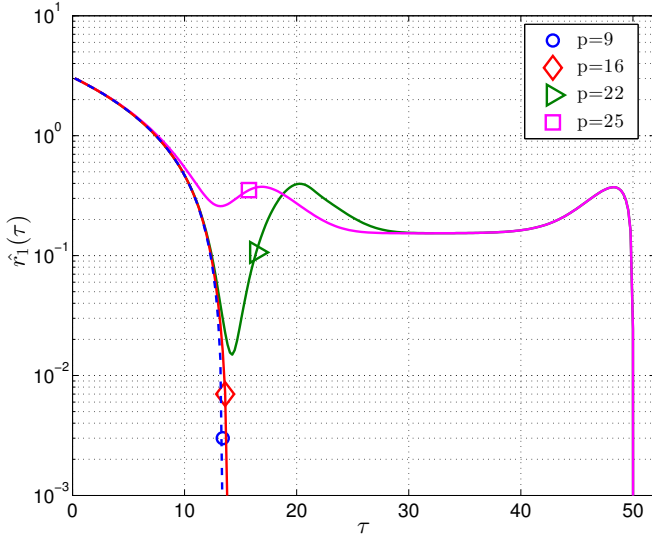


Fig. 5. For the ensemble $\tilde{\mathcal{C}}(3, 6, 50, p)$, we represent $\hat{r}_1(\tau)$ in (10) for $\epsilon = 0.47$ and $p = 9, p = 16, p = 22$ and $p = 25$.

techniques and results can easily be extended to general (l, r) -regular constructions. The $\mathcal{L}(3, 6, L)$ ensemble consists of two connected $(3, 6)$ -regular SC-LDPC code chains of length L . The low-degree check nodes at one end of the first chain are connected to variable nodes at interior positions of the second chain. We proceed similarly with the second chain, connecting check nodes at one end of the chain to interior positions in the first chain. The $\mathcal{L}(3, 6, L)$ ensemble for $L = 15$ is illustrated in Fig. 6. In similar fashion to the construction of the $\tilde{\mathcal{C}}(3, 6, L, p)$ ensemble, we use three consecutive positions to connect variable nodes of one chain to check nodes at the end of the other chain, creating variable nodes of degree four at these three positions². The loop ensemble can also be represented by its connectivity matrix:

$$\mathbf{T}_{\mathcal{L}(3,6,L)} = \begin{bmatrix} \mathbf{T}_{\mathcal{C}(3,6,L)} & \mathbf{L}_2 \\ \mathbf{L}_1 & \mathbf{T}_{\mathcal{C}(3,6,L)} \end{bmatrix}, \quad (14)$$

where \mathbf{L}_1 and \mathbf{L}_2 are $(L+2) \times (L+2)$ matrices that define the links between variable nodes in one chain and check nodes in the other chain. The dimension of \mathbf{T} is now $D = 2(L+2)$. From (2), we see that the rates of the $\mathcal{C}(3, 6, L)$ and $\mathcal{L}(3, 6, L)$ code ensembles are the same.

In Fig. 6, we observe that there are four regions of the graph where the code is stronger: two regions where variable nodes in two consecutive positions are connected to low-degree check nodes and two regions where there are variable nodes of degree four in three consecutive positions. These four regions divide each chain into an “outer segment” (from position 1 to the first position containing variable nodes of degree four) and an “inner segment” (from the end of the outer segment to the last position of the chain) (see Fig. 6).

As shown in [11], [12], the asymptotic performance of the

²In [11], different ways to connect the chains are compared, and it is shown that the resulting BP threshold is sensitive to the type of connection for small chain lengths L ; however, the change for moderate to large L is marginal.

Rate	Ensemble	$\epsilon_{\mathcal{L}(3,6,L)}$	Ensemble	$\epsilon_{\mathcal{C}(3,6,L)}$
0.375	$\mathcal{L}(3, 6, 8)$	0.5445	$\mathcal{C}(3, 6, 8)$	0.522
0.4	$\mathcal{L}(3, 6, 10)$	0.5323	$\mathcal{C}(3, 6, 10)$	0.508
0.4167	$\mathcal{L}(3, 6, 12)$	0.5237	$\mathcal{C}(3, 6, 12)$	0.495
0.4333	$\mathcal{L}(3, 6, 15)$	0.5105	$\mathcal{C}(3, 6, 15)$	0.489
0.4444	$\mathcal{L}(3, 6, 18)$	0.4989	$\mathcal{C}(3, 6, 18)$	0.488
0.46	$\mathcal{L}(3, 6, 25)$	0.488	$\mathcal{C}(3, 6, 18)$	0.488
0.48	$\mathcal{L}(3, 6, 50)$	0.488	$\mathcal{C}(3, 6, 18)$	0.488

TABLE II
BEC THRESHOLDS ϵ^* FOR SEVERAL SC-LDPC CONNECTED CHAIN LOOP ENSEMBLES $\mathcal{L}(3, 6, L)$ AND SINGLE CHAIN ENSEMBLES $\mathcal{C}(3, 6, L)$.

$\mathcal{L}(3, 6, L)$ ensemble is improved if the sides of the inner loop each have length $\lfloor 2L/3 \rfloor$ and the maximum check node degree is limited to 6, *i.e.*, variable nodes in one chain can only be connected to low degree check nodes in the other chain. These requirements are fulfilled if \mathbf{L}_1 has ones at positions $(1, \lceil L/3 \rceil)$, $(1, \lceil L/3 \rceil + 1)$, and $(2, \lceil L/3 \rceil + 2)$ and \mathbf{L}_2 has ones at positions $(L+1, \lfloor 2L/3 \rfloor + 1)$, $(L+2, \lfloor 2L/3 \rfloor + 2)$, and $(L+2, \lfloor 2L/3 \rfloor + 3)$. The BEC BP thresholds $\epsilon_{\mathcal{L}(3,6,L)}$ for different values of L are shown in Table II. The thresholds $\epsilon_{\mathcal{C}(3,6,L)}$ of the corresponding single chain SC-LDPC code ensembles of the same rate are given for comparison. It is observed that the thresholds of the connected chain ensembles are larger than the thresholds of the corresponding single chain ensembles for chain lengths up to $L = 18$.

An intuitive understanding of these results is as follows. For $L \leq 18$, the outer segments are quite short and, in addition, they have strong sub-codes at both ends. Hence, compared to the inner segments, which have double the length ($\lfloor 2L/3 \rfloor$ positions) and are only connected to variables of degree four, the outer segments are much better protected. Even when we are operating above the threshold, information encoded in the outer segments can be successfully recovered. For example, in Fig. 7 we plot the expected graph evolution of the normalized number of degree-one check nodes $\hat{r}_1(\tau)$ for the $\mathcal{L}(3, 6, 15)$ ensemble for different ϵ values above its BP threshold. Here, we plot $\hat{r}_1(\tau)$ against $v_{\text{outer}}(\tau)/L$, where

$$\hat{v}_{\text{outer}}(\tau) \doteq \sum_{i \in \text{outer segments}} \hat{v}_i(\tau) \quad (15)$$

is the expected fraction of undecoded bits at time τ that belong to the outer segments. We also include, for comparison, one curve corresponding to the $\mathcal{L}(3, 6, 50)$ ensemble. We observe that the outer segments can be decoded at channel erasure probabilities up to $\epsilon = 0.5365$, while the ensemble threshold is only $\epsilon_{\mathcal{L}(3,6,15)} = 0.5105$. Therefore, for $\epsilon \leq \epsilon_{\mathcal{L}(3,6,15)}$, the outer segments are almost surely decoded and, after reducing the graph accordingly, the two inner segments get essentially disconnected and low-degree check nodes are created at their ends. Consequently, the loop ensemble is decoded as two independent chains of length $2L/3$. Indeed, from Table II we observe that the loop thresholds $\epsilon_{\mathcal{L}(3,6,18)}$, $\epsilon_{\mathcal{L}(3,6,15)}$, and $\epsilon_{\mathcal{L}(3,6,12)}$ are only slightly larger than the thresholds of the single chain ensembles $\epsilon_{\mathcal{C}(3,6,12)}$, $\epsilon_{\mathcal{C}(3,6,10)}$, and $\epsilon_{\mathcal{C}(3,6,8)}$, respectively, with length $2L/3$.

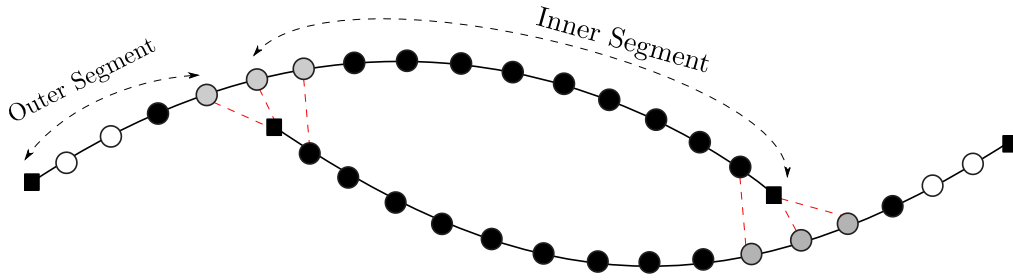


Fig. 6. Graphical representation of the $\mathcal{L}(3, 6, L)$ ensemble for $L = 15$. Dashed lines represent the additional edges added, gray dots indicate positions with variable nodes of degree four, and white dots positions with variable nodes connected to check nodes of degree two or four.

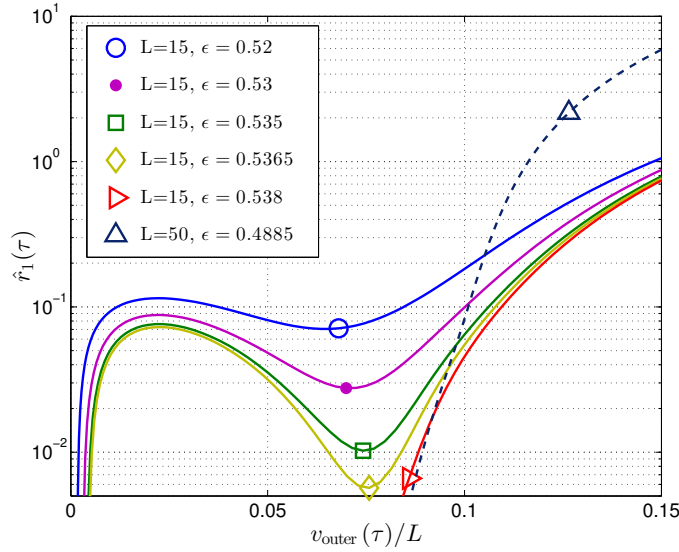


Fig. 7. Expected graph evolution of the normalized number of degree-one check nodes in the graph $\hat{r}_1(\tau)$ for the ensemble $\mathcal{L}(3, 6, L)$ as a function of $v_{\text{outer}}(\tau)$ (as defined in (15)) for different values of ϵ above the code threshold.

From the finite-length perspective, once the outer segments are decoded, the decoding of the two remaining shorter chains is described by the single critical-point model in (11), which is consistent with the single critical-point behavior observed as well for the $\mathcal{L}(3, 6, L)$ ensemble in [22]. For instance, in Fig. 8 we give expected graph evolution $\hat{r}_1(\tau)$ computed for the $\mathcal{L}(3, 6, L)$ ensemble with $L = 15$ and different values of ϵ , where we have also included the corresponding curves for the $\mathcal{C}(3, 6, 2L/3 = 10)$ ensemble³. For both ensembles we can observe a single critical point, but the evolution of the $r_1(\tau)$ process for the loop ensemble combines the contribution of the two connected chains. Therefore, its mean $\hat{r}_1(\tau)$ in Fig. 8, but also its variance, are higher than the corresponding mean and variance of the process $r_1(\tau)$ for the $\mathcal{C}(3, 6, 10)$ ensemble. By simulation, we have observed that the ratio of the mean $\hat{r}_1(\tau)$ to its standard deviation is approximately the same for both ensembles.

In conclusion, for short chain lengths $L = 5, \dots, 18$, the $\mathcal{L}(3, 6, L)$ loop ensemble has significantly better thresholds

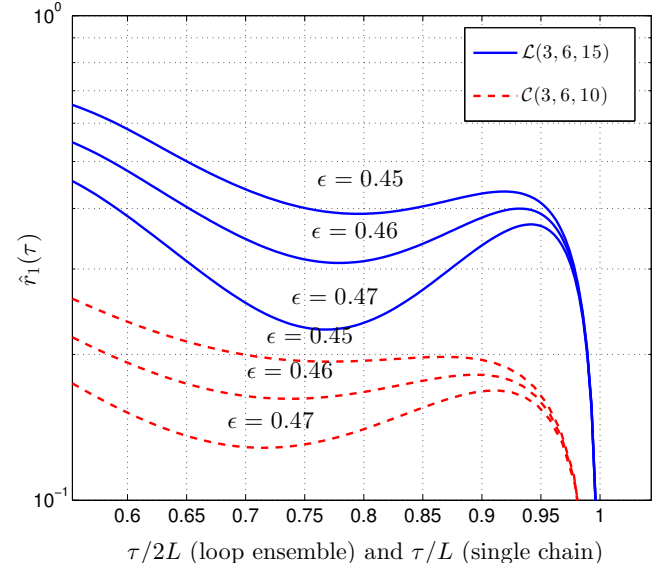


Fig. 8. Expected graph evolution of the fraction of degree one check nodes $\hat{r}_1(\tau)$ for the ensembles $\mathcal{L}(3, 6, 15)$ and $\mathcal{C}(3, 6, 10)$ with different values of ϵ .

than the $\mathcal{C}(3, 6, L)$ single chain ensemble, while maintaining similar finite-length scaling behavior, i.e., the single critical point model. These results explain the better finite-length block error performance of the loop ensemble compared to the single chain ensemble of the same length and rate. More specifically, both ensembles have the same chain length L , but the single chain has $2M$ bits per position compared to M bits per position in the case of the loop. This is demonstrated by simulation in Fig. 9 for the case $L = 15$, where we give simulated block error rate curves for the ensembles $\mathcal{L}(3, 6, 15)$ with 512 bits per position and $\mathcal{C}(3, 6, 15)$ with 1024 bits per position. In this figure, we also include the respective performance estimates for the two ensembles using the scaling law in (11).

The picture, however, is reversed when we consider the loop ensemble with large L values. For instance, in Fig. 9, the loop ensemble block error rate with $L = 50$ is clearly outperformed by the single chain ensemble of the same length and rate. For large L values, the (longer) outer segments do not display such strong performance above the ensemble threshold. For example, in Fig. 7 we include the $\hat{r}_1(\tau)$ vs. $v_{\text{outer}}(\tau)/L$ curve for the $\mathcal{L}(3, 6, 50)$ ensemble and ϵ slightly above the code

³ To assist visualization of the results, we have normalized the scaling of the horizontal axis by $2L$ in the loop case and by L in the single chain case.

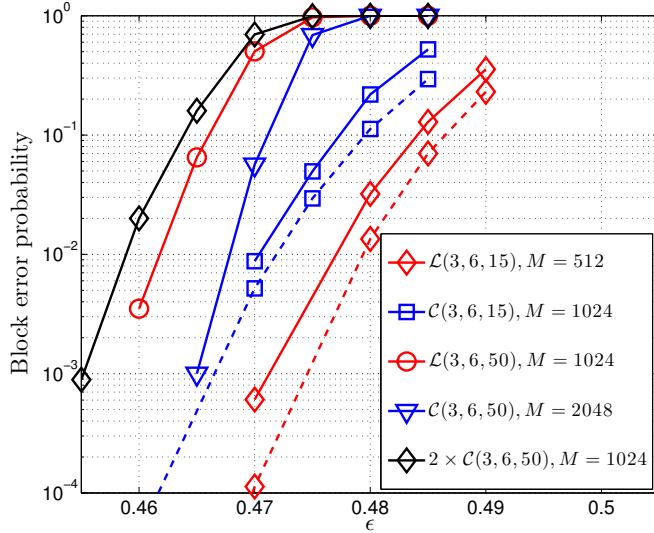


Fig. 9. Simulated block error rate performance for the ensembles $\mathcal{L}(3, 6, L)$ with 512 bits per position and $\mathcal{C}(3, 6, L)$ with 1024 bits per position for $L = 15$ and $L = 50$. In the $L = 15$ case, the performance approximation (dashed lines) obtained using the scaling law in (11) is included.

threshold $\epsilon_{\mathcal{L}(3,6,50)} \approx 0.48815$. Observe that the curve goes to zero for $v_{\text{outer}}(\tau) > 0$, which indicates that, by the time decoding fails, there is still a significant fraction of undecoded bits in the outer segment. For large values of L , the two regions in the graph with variable nodes of degree four are far apart from each other and from the two strong boundary positions with low degree check nodes. As demonstrated in Section III for the $\tilde{\mathcal{C}}(3, 6, 50, p)$ ensemble, when isolated, these regions are only effectively decoded for ϵ values much below $\epsilon_{\mathcal{L}(3,6,L)}$. In practice, this means a) there is no gain in threshold with respect to the $\mathcal{C}(3, 6, L)$ ensemble and b) that PD for the $\mathcal{L}(3, 6, L)$ ensemble is performed in a wave-like fashion from the boundary positions with low-degree check nodes towards the interior positions. Therefore, close to the threshold, there is no practical advantage with respect to an SC-LDPC code consisting of two independent long chains of length L and M bits per positions. Using (12), the block error probability of this ensemble can be estimated as follows:

$$P_{2 \times \mathcal{C}(l,r,L)} \approx 1 - \exp\left(-2 \frac{\epsilon L \bar{y}}{\mu_0(M, \epsilon, l, r)}\right) \quad (16)$$

and this error probability is much worse than for a single chain of the same rate, length L , and $2M$ bits per position [21], [22], since μ_0 in (12) grows exponentially in M . We demonstrate these results in Fig. 9, where we give simulated block error rates for the $\mathcal{L}(3, 6, 50)$ ensemble with 1024 bits per position and for the $\mathcal{C}(3, 6, 50)$ ensemble with 2048 bits per position. In addition, to show that, close to the threshold, the performance of the $\mathcal{L}(3, 6, 50)$ ensemble scales approximately as in (16), we also include the simulated block error performance of a code consisting of two independent $\mathcal{C}(3, 6, 50)$ chains with $M = 1024$ bits per position (\diamond).

To summarize, while we have seen that significant performance gains can be obtained by connecting short chains,

for long chains it is much better to increase the number of bits M per position in a single SC-LDPC chain rather than trying to obtain improvements by connecting multiple chains. Nevertheless, we will show in the next section that this result does not prevent us from finding SC-LDPC codes formed by connecting long chains that lead to substantial performance improvements.

V. A DIFFERENT APPROACH: CONNECTING CONSECUTIVE CHAINS

We start by considering the encoding and transmission of codewords using a single $\mathcal{C}(3, 6, L)$ code chain. The information stream is divided into blocks of $MLr_{\mathcal{C}(3,6,L)}$ bits, which are then independently encoded, transmitted, and decoded at the receiver. In the following, by transmitting independent chains we mean the transmission of independent codewords that belong to a $\mathcal{C}(3, 6, L)$ code chain. From (12), the probability that there is at least one decoding failure in N independent consecutive chains scales with M and L as:

$$P_{N \times \mathcal{C}(l,r,L)} \approx 1 - \exp\left(-N \frac{\epsilon L \bar{y}}{\mu_0(M, \epsilon, l, r)}\right). \quad (17)$$

In this section, we show that the performance of this system can be improved if we do not transmit independent chains, but instead data is encoded in a continuous fashion using a convolutional-like structure based on connected SC-LDPC code chains. We refer to such an encoding scheme as continuous chain (CC) transmission of SC-LDPC codes. More specifically, N consecutive SC-LDPC code chains, that without CC transmission are independently transmitted and decoded, will be now connected such that the overall robustness against failures is improved and the decoding complexity and decoding delay are not significantly affected.

In the following, we present some examples that illustrate the benefits of the CC coding/transmission scheme. We first propose some SC-LDPC code ensembles specifically designed for this application and then we discuss the feasibility of CC transmission.

A. SC-LDPC code ensembles for CC transmission

In previous sections we have seen that, when connecting SC-LDPC code chains, the local degree distribution around the connecting points plays an crucial role. In particular, the analysis of the $\tilde{\mathcal{C}}(3, 6, L, p)$ ensemble in Section III showed that the two regions formed by variables of degree four result in a decoding bottleneck unless they are located very close together, thus forming a single well protected graph region. We drew similar conclusions when analyzing the loop ensemble $\mathcal{L}(l, r, L)$ for large L , where the distance between connection points is so large that they do not really help to improve the decoding performance compared to an ensemble consisting of two single chains. In the light of these results, we now propose alternative structures based on connecting long chains that are suited for CC transmission. The main design difference is that we now break the symmetry of the loop ensemble by connecting two single chains such that one is more robust

against failures while the other maintains approximately the same decoding properties as a single chain.

This is precisely the effect achieved by the ensemble illustrated in Fig. 10. This ensemble consists of two layers with one single chain of length L per layer, and it is denoted by $\mathcal{S}(3,6,L,2)$. In general, we denote such ensembles as $\mathcal{S}(3,6,L,N)$, where N stands for the number of layers. The layer j chain is simply referred to as chain j . In Fig. 10, the chains in the two layers have been connected to increase the protection of the variable nodes in the four middle positions of chain 1, from position $\lceil L/2 \rceil$ to position $\lceil L/2 \rceil + 3$, by adding the following edges:

- One edge from each variable node at position $\lceil L/2 \rceil$ in chain 1 to a check node at position 1 in chain 2.
- Two edges from each variable node at position $\lceil L/2 \rceil + 1$ in chain 1 to check nodes at positions 1 and $(L+1)$ in chain 2.
- Two edges from each variable node at position $\lceil L/2 \rceil + 2$ in chain 1 to check nodes at positions 2 and $(L+2)$ in chain 2.
- One edge from each variable node at position $\lceil L/2 \rceil + 3$ in chain 1 to a check node at position $(L+2)$ in chain 2.

The connectivity matrix of this ensemble is given by

$$\mathbf{T}_{\mathcal{S}(3,6,L,2)} = \begin{bmatrix} \mathbf{T}_{\mathcal{C}(3,6,L)} & \mathbf{0} \\ \mathbf{CCT} & \mathbf{T}_{\mathcal{C}(3,6,L)} \end{bmatrix}, \quad (18)$$

where \mathbf{CCT} is an $(L+2) \times (L+2)$ matrix with ones at positions $(1, \lceil L/2 \rceil)$, $(1, \lceil L/2 \rceil + 1)$, and $(2, \lceil L/2 \rceil + 2)$, which corresponds to the connections between the left end of the second chain and the middle positions of the first chain, and at positions $(L+1, \lceil L/2 \rceil + 1)$, $(L+2, \lceil L/2 \rceil + 2)$, and $(L+2, \lceil L/2 \rceil + 3)$, which corresponds to the connections between the right end of the second chain and the middle positions of the first chain. In this way, we create a region with much better protection in the middle of chain 1, where the variable degree profile is $(4, 5, 5, 4)$, and all check nodes in chain 2 now have degree 6. Note that, while chain 1 contains three strong sub-codes, all variable nodes in chain 2 are of degree 3 and all check nodes are of degree 6. Therefore, to be effectively decoded, chain 2 relies on the decoding of the four middle positions of chain 1, where the variable nodes are better protected. Note that, if positions $\lceil L/2 \rceil, \dots, \lceil L/2 \rceil + 3$ in chain 1 are correctly decoded, then chain 2 has the same DD as the SC-LDPC single chain discussed in Section II. Decoding chain 2 is then similar to decoding the $\tilde{\mathcal{C}}(3,6,L, \lceil L/2 \rceil)$ ensemble discussed in Section III, i.e., to initiate wave-like decoding towards the interior positions, decoding relies on the decoding of a single protected graph region where all check nodes are of degree 6 and the variable nodes are of degrees $(4, 5, 5, 4)$.

By computing the expected evolution of degree-one check nodes in the graph for the $\mathcal{S}(3,6,L,2)$ ensemble, we observe that, despite the fact that the $\mathcal{S}(3,6,L,2)$ threshold is exactly the same as for the single SC-LDPC code chain ($\epsilon_{\mathcal{S}(3,6,L,2)} \approx 0.48815$), all variable nodes in positions $(\lceil L/2 \rceil, \dots, \lceil L/2 \rceil + 3)$ in chain 1 can be successfully decoded up to an erasure probability of $\epsilon = 0.505$. Therefore, for $\epsilon \leq \epsilon_{\mathcal{S}(3,6,L,2)}$, these positions are almost surely decoded,

which has two important consequences. First, the finite-length error rate in chain 2 will be similar to that of the single SC-LDPC code chain and, second, low-degree check nodes will be created at positions $\lceil L/2 \rceil - 1, \lceil L/2 \rceil, \lceil L/2 \rceil + 3$, and $\lceil L/2 \rceil + 4$ in chain 1. Consequently, this chain is essentially decoded as two interdependent chains of length $\lceil L/2 \rceil$. Further, if we use intermediate chain lengths, i.e., $L \in [40, 70]$, the decoding of each of the two segments of length $\lceil L/2 \rceil$ is well predicted by the single-critical point model in (11), and we can benefit from much better properties in terms of finite-length scaling behavior, as discussed in Section II-C. As a consequence, the block error rate of chain 1 will be better than for a single SC-LDPC code chain of length L . These statements will be further confirmed by simulation in Section V-C.

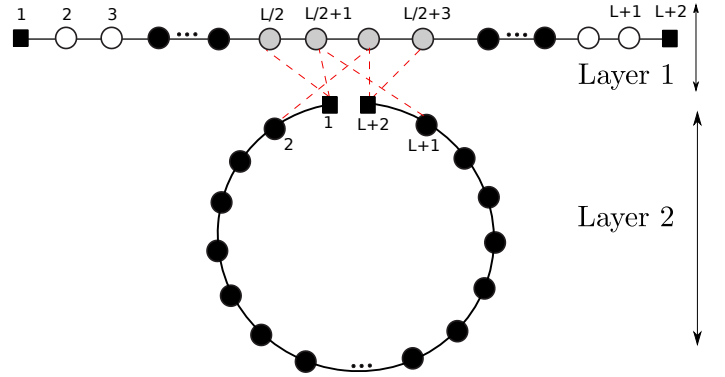


Fig. 10. Graphical representation of the $\mathcal{S}(3,6,L,2)$ ensemble in (18).

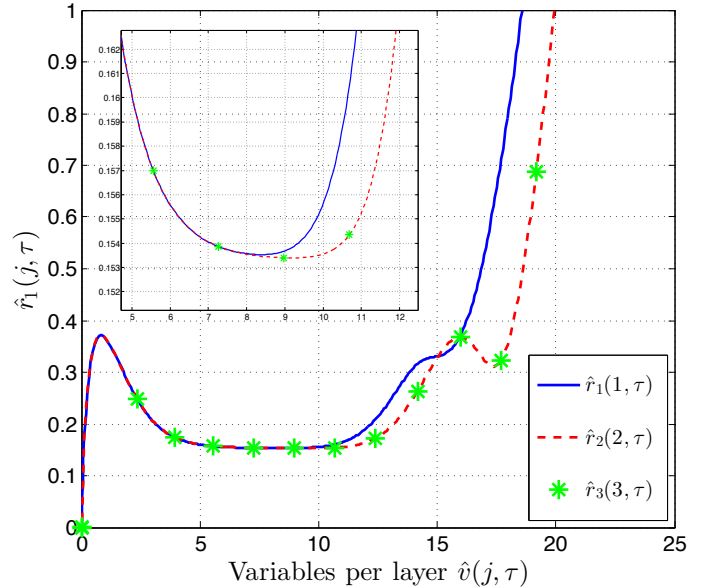
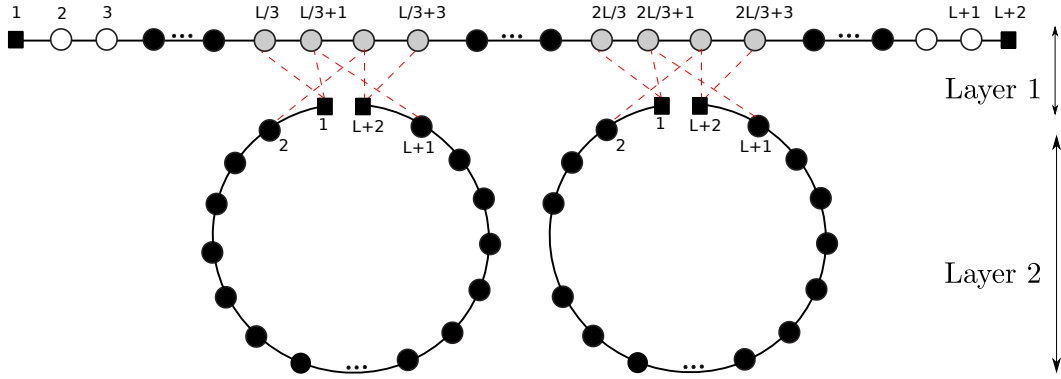


Fig. 11. Evolution of the normalized number of degree-one check nodes per layer $\hat{r}_1(j, \tau)$, $j = 1, 2, 3$, as a function of the normalized number of variable nodes per layer $\hat{v}(j, \tau)$ during PD for the $\mathcal{S}(3,6,L,4)$ ensemble with $\epsilon = 0.45$.

The design methodology used to construct the $\mathcal{S}(3,6,L,2)$ ensemble can be extended to an arbitrary number N of layers, and the resulting ensemble has the same rate as the

Fig. 12. Graphical representation of the $\mathcal{S}(3, 6, L, 2, 2)$ ensemble.

$\mathcal{C}(3, 6, L)$ ensemble for any number of layers. For instance, the connectivity matrix of the $\mathcal{S}(3, 6, L, 3)$ ensemble is given by

$$\mathbf{T}_{\mathcal{S}(3,6,L,3)} = \begin{bmatrix} \mathbf{T}_{\mathcal{C}(3,6,L)} & \mathbf{0} & \mathbf{0} \\ \mathbf{CCT} & \mathbf{T}_{\mathcal{C}(3,6,L)} & \mathbf{0} \\ \mathbf{0} & \mathbf{CCT} & \mathbf{T}_{\mathcal{C}(3,6,L)} \end{bmatrix}, \quad (19)$$

where the **CCT** matrix is of the same form as in (18). By using the low-degree check nodes of chain 3, we increase the protection of the variable nodes at positions $\lceil L/2 \rceil, \dots, \lceil L/2 \rceil + 3$ in chain 2, which yields a performance improvement similar to that obtained in chain 1. More specifically, once the positions in the middle of the chain 1 have been decoded, besides splitting this chain into two sub-chains of length $\lceil L/2 \rceil$, low-degree check nodes are created at the boundary positions of chain 2. This chain now looks similar to chain 1 at the beginning of the decoding process and, consequently, the performance in layers 1 and 2 is approximately the same. Alternatively, for an arbitrary number N of layers, the finite length performance per chain can be analyzed by computing the $\mathcal{S}(3, 6, L, N)$ mean graph evolution during PD, as discussed in Section II-B. For $N = 4$, in Fig. 11 we plot the expected graph evolution of the normalized number of degree-one check nodes per layer $\hat{r}_1(j, \tau)$, $j = 1, \dots, N - 1$, as a function of the normalized number of variable nodes per layer $\hat{v}(j, \tau)$. As we can observe from the magnification in the left upper corner, the finite-length performance of chains 1 to 3 is determined by a single critical point of approximately the same height, which indicates a similar exposure to decoding failures. We will confirm this result in Section V-C.

At this point, it is important to clarify that, even though N consecutive chains are now linked in the $\mathcal{S}(3, 6, L, N)$ ensemble, a decoding failure in a particular chain does not prevent the rest of chains from being successfully decoded. As discussed above for the $\mathcal{S}(3, 6, L, 2)$ ensemble, the highly protected region created in the middle of each chain is decoded almost surely at any erasure below the $\mathcal{S}(3, 6, L, N)$ threshold, and this means that decoding failures at chain j do not propagate to lower chains. Therefore, the ratio of the number of strongly protected chains to the total number of chains is given by $\eta_{\mathcal{S}(3,6,L,N)} = \frac{N-1}{N}$.

The key idea behind the performance improvement experienced by the different chains is to break them into chains of

shorter lengths for which the finite-length scaling behavior is more robust. When the rate constraints are even stricter and, as a consequence, we must use even longer chains, *e.g.*, $L = 100$, the $\mathcal{S}(3, 6, L, N)$ ensemble might not be able to provide the same gain per layer as for a shorter chain, *e.g.*, $L = 50$. In this case, we can break each chain of length $L = 100$ into three shorter segments by creating two intermediate regions with stronger protection. We illustrate this construction for a two-layer case in Fig. 12, and we denote such ensembles by $\mathcal{S}(3, 6, L, N, t)$, where t refers to the number of chains in each layer that are used to improve the decoding of each chain in the above layer⁴. In this case $t = 2$.

Note that, if we add another layer to the ensemble in Fig. 12, we will need four chains to improve the performance of the chains in layers 1 and 2. For an arbitrary number N of layers, the ratio of the number of strongly protected chains to the total number of chains is:

$$\eta_{\mathcal{S}(3,6,L,N,t)} = \frac{\sum_{j=1}^{N-1} t^{j-1}}{N}, \quad (20)$$

which tends to $1/t$ with increasing N . For the ensemble $\mathcal{S}(3, 6, L, N, 2)$, this implies that, when using CC transmission, half of the chains will enjoy better performance with no significant increase in encoding/decoding complexity, as we explain in the next section.

B. Feasibility of CC transmission

Without loss of generality, in this section we consider CC transmission using the $\mathcal{S}(3, 6, L, N)$ ensemble described above to show that, compared to transmission of independent chain codewords, CC transmission using connected SC-LDPC code chains only requires some additional memory and a different transmission order for the encoded bits.

1) *Encoding:* For a given code belonging the $\mathcal{C}(3, 6, L)$ ensemble, the encoding process can be implemented sequentially using the syndrome former encoder proposed in [30].

⁴The ensemble $\mathcal{S}(3, 6, L, N, t)$ for $t = 1$ is the one discussed above and simply denoted by $\mathcal{S}(3, 6, L, N)$.

Let $\mathbf{u}^{(i)}$, $i = 2, \dots, L+1$, be a sub-block of $M\mathbf{r}_{C(3,6,L)}$ information bits and $\mathbf{v}^{(i)}$ the corresponding sub-block of M encoded bits. Using the syndrome former encoder, to compute $\mathbf{v}^{(i)}$ we only need $\mathbf{u}^{(i)}$ and the previously encoded blocks $\mathbf{v}^{(i-1)}, \mathbf{v}^{(i-2)}, \dots$, corresponding to the positions in the SC-LDPC code chain whose variable nodes are connected to the encoded bits $\mathbf{v}^{(i)}$ by the parity check nodes at position i . For example, according to the first two rows of the connectivity matrix in (1), $\mathbf{v}^{(1)}$ can be obtained directly from $\mathbf{u}^{(1)}$ and, given $\mathbf{v}^{(1)}$ and $\mathbf{u}^{(2)}$, we can compute $\mathbf{v}^{(2)}$.

Using the syndrome former encoder, the process of encoding N consecutive layers of the $\mathcal{S}(3,6,L,N)$ ensemble is essentially equivalent in complexity to encoding N consecutive but independent codewords of the single chain ensemble. The only difference is that, after the first chain in Fig. 10 has been encoded, the encoding of the first sub-block of the second chain, viz. $\mathbf{v}_2^{(1)}$, requires $\mathbf{u}_2^{(1)}$ but also $\mathbf{v}_1^{(\lceil L/2 \rceil)}, \mathbf{v}_1^{(\lceil L/2 \rceil + 1)}$, and $\mathbf{v}_1^{(\lceil L/2 \rceil + 2)}$. Similarly, to compute the last encoded sub-block in the second chain, viz. $\mathbf{v}_2^{(L+1)}$, we also need $\mathbf{v}_1^{(\lceil L/2 \rceil + 1)}, \mathbf{v}_1^{(\lceil L/2 \rceil + 2)}$, and $\mathbf{v}_1^{(\lceil L/2 \rceil + 3)}$. Therefore, the encoding process has the same complexity as encoding N independent chain codewords, but we have some additional memory requirements to store the encoded sub-blocks $\mathbf{v}_j^{(\lceil L/2 \rceil)}, \mathbf{v}_j^{(\lceil L/2 \rceil + 1)}, \mathbf{v}_j^{(\lceil L/2 \rceil + 2)}$, and $\mathbf{v}_j^{(\lceil L/2 \rceil + 3)}$ that are necessary to encode the chain at layer $j+1$, $j = 1, \dots, N-1$.

2) *Window decoding and transmission order*: Efficient decoding of long SC-LDPC code chains with low decoding delay is based on windowed BP decoding [6], [23]. In a nutshell, decoding is restricted to a window of W positions that ‘slides’ over the graph, exploiting the convolutional structure of the SC-LDPC code parity check matrix: as bits in the left most positions of the window are decoded, the window is shifted right and new bits are included in the decoding window (see [6] and [23] for further details). For sufficiently large W , e.g., a window of length $W = 12$ positions for the standard $\mathcal{C}(3,6,L)$ SC-LDPC code chain described in Section II, the performance is indistinguishable from a standard BP decoder, while the delay is much less, since decoding can be initiated before receiving the entire codeword.

The same decoding principle can be simply adapted to perform efficient decoding of CC transmission of SC-LDPC codes. For example, for the $\mathcal{S}(3,6,L,N)$ ensemble with $N = 2$ in Fig. 10, the window decoder can be initiated at the first position of chain 1. The window will shift until it reaches the middle positions of the chain, whose bits are better protected since they are also connected to check nodes at the boundary positions of chain 2. Therefore, to efficiently continue window decoding of chain 1, channel information from the variable nodes at positions 2 and $(L+1)$ of chain 2 must be available. Note that we need this information even before receiving channel information from the variable nodes at the remaining positions of chain 1. Once chain 1 has been decoded, window decoding of chain 2 can be started using the information already available from variable node positions 2 and $L+1$.

Therefore, implementing efficient window decoding for CC

transmission reduces to a change in the order in which the encoded bits are transmitted, so that the receiver can have the necessary information at the appropriate time. Clearly, it is necessary for both the transmitter and the receiver to be aware of the transmission order. Returning to the 2-layer example in Fig. 10, it would be sufficient to transmit the encoded blocks $\mathbf{v}_j^{(i)}$ for $j = 1, 2$ and $i = 2, \dots, L+1$ in the following order:

$$\begin{aligned} \mathbf{v}_1^{(1)} \rightarrow \mathbf{v}_1^{(2)} \rightarrow \dots \rightarrow \mathbf{v}_1^{(\lceil L/2 \rceil + 3)} \rightarrow \mathbf{v}_2^{(1)} \rightarrow \mathbf{v}_2^{(L+1)} \\ \rightarrow \mathbf{v}_1^{(\lceil L/2 \rceil + 4)} \rightarrow \dots \rightarrow \mathbf{v}_1^{(L+1)} \rightarrow \mathbf{v}_2^{(3)} \rightarrow \dots \rightarrow \mathbf{v}_2^{(L)}. \end{aligned}$$

If we add one layer to the example in Fig. 10, i.e., we use the $\mathcal{S}(3,6,L,N)$ ensemble with $N = 3$ for CC transmission, we would use the same transmission policy between the boundary positions in chain 3, i.e., $\mathbf{v}_3^{(1)}$ and $\mathbf{v}_3^{(L+1)}$, and the positions $\mathbf{v}_2^{(\lceil L/2 \rceil + 3)}, \dots, \mathbf{v}_2^{(\lceil L/2 \rceil + 1)}$ in chain 2.

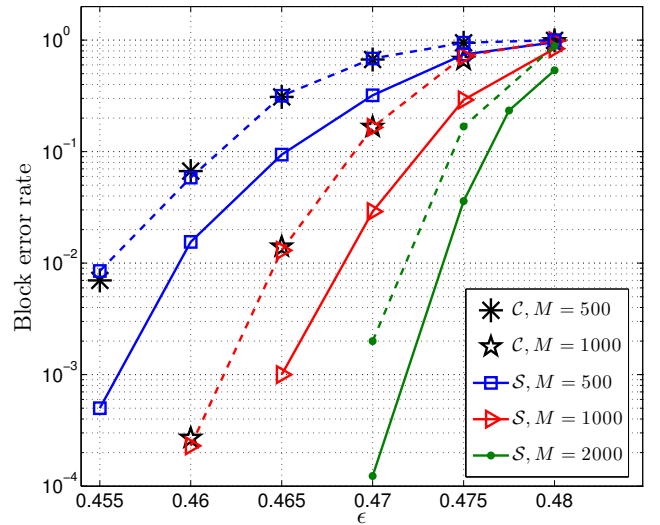


Fig. 13. Block error rate computed for the chains in layer 1 (solid lines) and layer 2 (dashed lines) of a code drawn from the $\mathcal{S}(3,6,L=50,N=2)$ ensemble. The performance of representative members of the $\mathcal{C}(3,6,50)$ ensemble is also shown for comparison.

C. Simulation results

Finally, we include simulation results that corroborate our analysis of the ensembles presented in Section V-A for CC transmission of SC-LDPC codes. The codes are constructed so that they do not contain cycles of length four. Performance curves are obtained after averaging 10^4 simulations.

We start by plotting performance results for a code drawn from the $\mathcal{S}(3,6,L=50,N=2)$ ensemble⁵ over the BEC. In Fig. 13, we show the block error rate computed for chain 1 (solid lines) and chain 2 (dashed lines), as well as the performance of representative members of the ensemble $\mathcal{C}(3,6,50)$. As predicted, we obtain a significant gain in

⁵All codes used to generate the simulation results in this section are accessible at the first author's web site: <http://www.tsc.uc3m.es/~olmos/>.

performance for chain 1, of almost one order of magnitude, even for codes with 2000 bits per position. Also, we note that there is no performance degradation in chain 2 with respect to a single unconnected chain⁶. This confirms a) that our design criterion based on creating a single powerful connection point is appropriate, and b) that the modified chain ensemble $\mathcal{C}(3, 6, L, p)$ shown in Fig. 4 for $p = \lceil L/2 \rceil$ provides the same performance as the standard chain ensemble.

In Fig. 14, we show results for the ensemble $\mathcal{S}(3, 6, 50, N)$ with three layers, i.e., $N = 3$. The performance of the first two layers is plotted in solid lines, while the performance of chain 3 is given in dashed lines. In addition to what we observed in Fig. 13, note that the performance measured for chains 1 and 2 is essentially the same, and hence there is no degradation in performance in the second layer with respect to the first layer, even though it does not have low-degree check nodes at each end. This is critical to the success of the CC transmission approach for an arbitrary number of layers, and this behavior has further been confirmed by simulation for ensembles with a larger number of layers. We omit the simulation results for the $\mathcal{S}(3, 6, L = 100, N = 2, t = 2)$ ensemble illustrated in Fig. 12, but, as expected, we have again observed that the performance of chain 1 is greatly improved, while the two chains in layer 2 provide similar performance to the SC-LDPC single chain ensemble $\mathcal{C}(3, 6, 100)$.

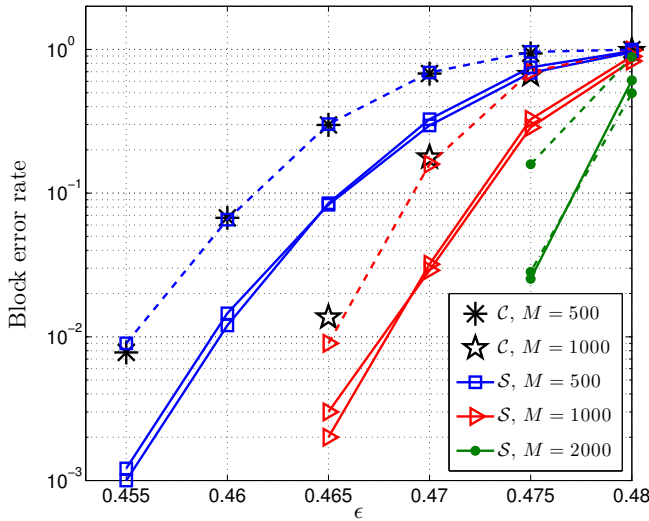


Fig. 14. Block error rate computed for the chains in layers 1 and 2 (solid lines) and layer 3 (dashed lines) of a code drawn from the $\mathcal{S}(3, 6, L = 50, N = 3)$ ensemble. The performance of representative members of the $\mathcal{C}(3, 6, 50)$ ensemble is also shown for comparison.

Finally, we want to emphasize that, though the finite-length analysis of the different ensembles was restricted to the BEC, our design and conclusions extend to other channels as well. In Fig. 15, we show results for the $\mathcal{S}(3, 6, 50, N)$ ensemble with $N = 2$ used for transmission over the BIAWGN channel.

⁶Note also that the behavior shown in Fig. 13 give us an *unequal error protection capability*, i.e., chain 1 is better protected than chain 2, that can be useful if we have different classes of data.

We represent the performance of chain 2 in dashed lines and the performance of chain 1 in solid lines. We also include performance results for the single chain ensemble $\mathcal{C}(3, 6, 50)$. As we can observe, the performance of chain 1 is substantially improved with respect to chain 2, which in turn has the same performance as the single chain of the same length and rate. In Fig. 16, we show simulated performance results over the BIAWGN channel for the $\mathcal{S}(3, 6, 50, N)$ ensemble with $N = 3$ and $M = 500$ bits per position. The solid lines represent the performance of the chains in the first two layers. As discussed above for the BEC, the performance in the first two layers of the $\mathcal{S}(3, 6, 50, 3)$ ensemble is not degraded, and we achieve performance gains with respect a single SC-LDPC code chain.

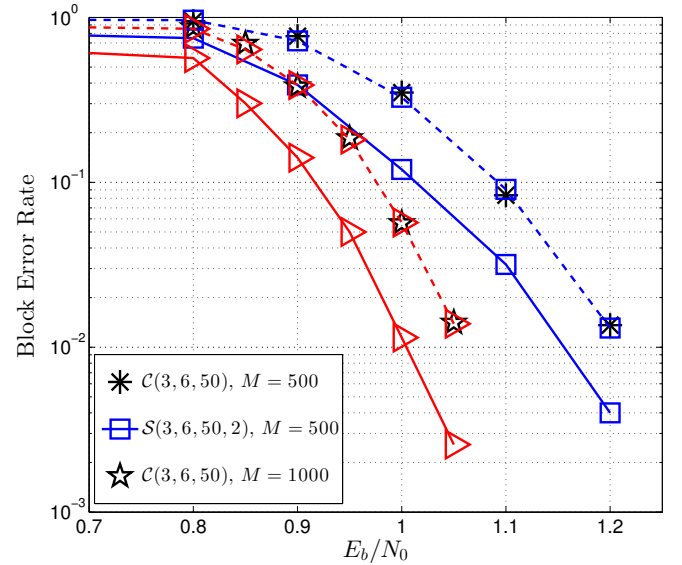


Fig. 15. Block error rate computed for the chains in layer 1 (solid lines) and layer 2 (dashed lines) of a code drawn from the $\mathcal{S}(3, 6, L = 50, N = 2)$ ensemble, used for transmission over the BIAWGN channel. The performance of representative members of the $\mathcal{C}(3, 6, 50)$ ensemble is also shown for comparison.

VI. CONCLUSIONS

In this paper, we have jointly analyzed SC-LDPC codes based on a single chain and codes based on connected chains from a finite-length perspective. First, we provided a unified vision of existing models, capturing the main scaling behavior between the block error rate and code parameters for a SC-LDPC code chain. As a novel contribution, we showed that there is an important design aspect regarding the robustness against failures of short SC-LDPC code chains compared to long chains. Ensembles based on two connected chains of the same length were proposed in [11], [12], where it was shown they are able to achieve thresholds closer to capacity. In this paper, we have shown that this gain in threshold is a direct consequence of a) the presence of regions in the graph where variables nodes are better protected, and b) their proximity to one another. Connected chains are shown to be particularly effective when these regions are strong enough to effectively break the chains into shorter, better protected

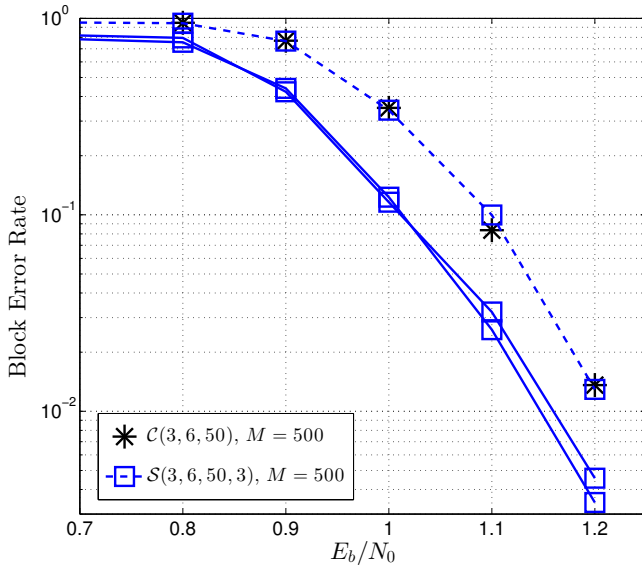


Fig. 16. Block error rate computed for the chains in layers 1 and 2 (solid lines) and layer 3 (dashed lines) of a code drawn from the $\mathcal{S}(3, 6, L = 50, N = 3)$ ensemble, used for transmission over the BIAWGN channel. The performance of a representative member of the $\mathcal{C}(3, 6, 50)$ ensemble is also shown for comparison.

chains. In particular, we have shown the loop ensemble created from two chains of length L essentially behaves, both in threshold and finite-length scaling behavior, as a single chain of shorter length. As a third contribution of the paper, a novel transmission scheme (CC transmission) designed to boost the performance of a system using long SC-LDPC code chains was introduced. We have shown that by connecting consecutive long chains, rather than transmitting the code-words corresponding to each one independently, we can obtain performance improvements for both the BIAWGN and BEC channels. Moreover, we have shown that CC transmission only requires minor changes to the order in which coded bits are transmitted and some additional memory requirements at the encoder. No significant increase in encoding/decoding complexity or decoding delay is necessary.

There are many possible variations on the construction of connected chain SC-LDPC code ensembles presented in this paper. By analyzing structurally simple ensembles, i.e., random ensembles instead of protograph-based ensembles, we have been able to provide insight into some critical questions regarding the relation between the structural parameters of connected chain SC-LDPC codes and their decoding performance. Finally, note that the principle of CC transmission is very general and may give rise to other novel encoding/transmission schemes.

REFERENCES

- [1] R. G. Gallager, "Low-density parity-check codes," Ph.D. dissertation, Massachusetts Institute of Technology, Cambridge, MA, 1963.
- [2] D. J. C. MacKay, "Good error-correcting codes based on very sparse matrices," *IEEE Transactions on Information Theory*, vol. 45, no. 2, pp. 399–431, Mar. 1999.
- [3] —, "Gallager codes that are better than turbo codes," in *Proc. Allerton Conference on Communication, Control, and Computing*, Monticello, IL, Sept. 1998.
- [4] C. Measson, A. Montanari, and R. Urbanke, "Maxwell construction: The hidden bridge between iterative and maximum a posteriori decoding," *IEEE Transactions on Information Theory*, vol. 54, no. 12, pp. 5277–5307, Dec. 2008.
- [5] S. Kudekar, T. Richardson, and R. Urbanke, "Spatially Coupled Ensembles Universally Achieve Capacity under Belief Propagation," *IEEE Transactions on Information Theory*, vol. 59, no. 12, pp. 7761–7813, Dec. 2013.
- [6] M. Lentmaier, A. Sridharan, D. J. Costello, Jr., and K. Sh. Zigangirov, "Iterative decoding threshold analysis for LDPC convolutional codes," *IEEE Transactions on Information Theory*, vol. 56, no. 10, pp. 5274–5289, Oct. 2010.
- [7] S. Kudekar, T. J. Richardson, and R. L. Urbanke, "Threshold saturation via spatial coupling: why convolutional LDPC ensembles perform so well over the BEC," *IEEE Transactions on Information Theory*, vol. 57, no. 2, pp. 803–834, Feb. 2011.
- [8] S. Kudekar and H. D. Pfister, "The effect of spatial coupling on compressive sensing," in *Proc. Allerton Conference on Communications, Control, and Computing*, Monticello, IL, Sept. 2010.
- [9] D. Truhachev, "Achieving awgn multiple access channel capacity with spatial graph coupling," *IEEE Communications Letters*, vol. 16, no. 5, pp. 585–588, May 2012.
- [10] K. Takeuchi, T. Tanaka, and T. Kawabata, "Improvement of BP-based CDMA multiuser detection by spatial coupling," in *Proc. IEEE International Symposium on Information Theory*, St. Petersburg, Russia, Aug. 2011.
- [11] D. Truhachev, D. G. M. Mitchell, M. Lentmaier, and D. J. Costello, Jr., "New Codes on Graphs Constructed by Connecting Spatially Coupled Chains," Submitted to *IEEE Transactions on Information Theory*, Dec. 2013. [Online]. Available: <http://arxiv.org/abs/1312.3368>
- [12] —, "Connecting spatially coupled LDPC code chains," in *Proc. IEEE International Conference on Communications (ICC)*, Ottawa, Canada, June 2012, pp. 2176–2180.
- [13] M. Lentmaier, D. G. M. Mitchell, G. P. Fettweis, and D. J. Costello, Jr., "Asymptotically good LDPC convolutional codes with AWGN channel thresholds close to the Shannon limit," in *Proc. 6th International Symposium on Turbo Codes and Iterative Information Processing*, Brest, France, Sept. 2010.
- [14] A. Amraoui, A. Montanari, T. Richardson, and R. Urbanke, "Finite-length scaling for iteratively decoded LDPC ensembles," *IEEE Transactions on Information Theory*, vol. 55, no. 2, pp. 473–498, Feb. 2009.
- [15] T. Nozaki, K. Kasai, and K. Sakaniwa, "Analytical Solution of Co-variance Evolution for Irregular LDPC Codes," *IEEE Transactions on Information Theory*, vol. 58, no. 7, pp. 4770–4780, July 2012.
- [16] J. Ezri, A. Montanari, and R. Urbanke, "A generalization of the finite-length scaling approach beyond the BEC," in *Proc. IEEE International Symposium on Information Theory*, Nice, France, June 2007, pp. 1011–1015.
- [17] T. J. Richardson and R. Urbanke, *Modern Coding Theory*. Cambridge University Press, Mar. 2008.
- [18] M. Luby, M. Mitzenmacher, M. Shokrollahi, and D. Spielman, "Efficient erasure correcting codes," *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 569–584, Feb. 2001.
- [19] A. Amraoui, R. Urbanke, and A. Montanari, "Finite-length scaling of irregular LDPC code ensembles," in *Proc. IEEE Information Theory Workshop (ITW)*, Rotorua, New Zealand, Aug. 2005.
- [20] P. M. Olmos and R. Urbanke, "A closed form scaling law for Spatially Coupled LDPC codes over the BEC," in *Proc. IEEE Information Theory Workshop (ITW)*, Sevilla, Spain, Sept. 2013.
- [21] —, "A scaling law to predict the finite-length performance of Spatially Coupled LDPC Codes." To be submitted to *IEEE Transactions on Information Theory*, 2014.
- [22] P. M. Olmos, D. T. David Mitchell, and D. J. Costello, Jr., "A finite length performance analysis of LDPC codes constructed by connecting spatially coupled chains," in *Proc. IEEE Information Theory Workshop (ITW)*, Sevilla, Spain, Sept. 2013.
- [23] A. Iyengar, M. Papaleo, P. Siegel, J. Wolf, A. Vanelli-Coralli, and G. Corazza, "Windowed decoding of protograph-based LDPC convolutional codes over erasure channels," *IEEE Transactions on Information Theory*, vol. 58, no. 4, pp. 2303–2320, April 2011.
- [24] P. M. Olmos and R. Urbanke, "Scaling behavior of Convolutional LDPC ensembles over the BEC," in *Proc. IEEE International Symposium on Information Theory (ISIT)*, St. Petersburg, Russia, Aug. 2011, pp. 1816–1820.
- [25] G. Kramer, M. Gastpar, and P. Gupta, "Cooperative strategies and capacity theorems for relay networks," *IEEE Transactions on Information Theory*, vol. 51, no. 9, pp. 3037–3063, Sept. 2005.

- [26] T. Cover and A. Gamal, "Capacity theorems for the relay channel," *IEEE Transactions on Information Theory*, vol. 25, no. 5, pp. 572–584, Sept. 1979.
- [27] J. Thorpe, "Low-density parity-check (LDPC) codes constructed from protographs," Jet Propulsion Laboratory, Pasadena, CA, INP Progress Report 42-154, Aug. 2003.
- [28] M. Lentmaier, D. Mitchell, G. Fettweis, and D. J. Costello, Jr., "Asymptotically regular LDPC codes with linear distance growth and thresholds close to capacity," in *Proc. Information Theory and Applications (ITA) Workshop, San Diego, CA*, Feb. 2010, pp. 1–8.
- [29] D. Mitchell, A. Pusane, M. Lentmaier, and D. J. Costello, Jr., "Exact free distance and trapping set growth rates for LDPC convolutional codes," in *Proc. IEEE International Symposium on Information Theory (ISIT), St. Petersburg, Russia*, Aug. 2011, pp. 1096–1100.
- [30] A. E. Pusane, A. Jiménez Felström, A. Sridharan, M. Lentmaier, K. Sh. Zigangirov, and D. J. Costello, Jr., "Implementation aspects of LDPC convolutional codes," *IEEE Transactions on Communications*, vol. 56, no. 7, pp. 1060–1069, July 2008.

APPENDIX A

EXPECTED EVOLUTION IN ONE PD STEP

Here we describe the computation of the graph expectations in (8) and (9). Assume the graph at iteration ℓ has DD $\{R_{j,u}(\ell), V_u(\ell)\}$ for $u \in [1, D]$ and $j = 1, \dots, r$, and the position where we remove a degree-one check node is denoted as $\text{pos}(\ell)$. At this position, the actual check node removed is chosen with uniform probability, and the probability $P(\text{pos}(\ell) = u) \doteq p_u(\ell)$ is given by

$$p_u(\ell) = \frac{R_{1,u}(\ell)}{\sum_{i=1}^D R_{1,i}(\ell)}. \quad (21)$$

In the following, let $m = \text{pos}(\ell)$. The variable node connected to the degree-one check node removed is at position u with probability

$$\lambda_{m,u}(\ell) = \frac{V_u(\ell)T_{m,u}}{\sum_{i=1}^D V_i(\ell)T_{m,i}}, \quad (22)$$

where the denominator represents the total number of variable nodes connected to check nodes at position m . Therefore a check node at position u loses one edge with probability

$$\xi_{m,u}(\ell) = \sum_{q=1}^D T_{u,q} \lambda_{m,q}(\ell). \quad (23)$$

The expected DD evolution at position m is easy to compute, since we just remove an edge of right degree one. Thus $\mathbb{E}[R_{j,m}(\ell+1) - R_{j,m}(\ell) | \text{pos}(\ell) = m]$ is equal to -1 for $j = 1$ and zero otherwise. For $u \neq m$, the graph loses one edge with probability $\xi_{m,u}(\ell)$, given by (23). This lost edge was connected to a degree- j check node with probability

$$\frac{R_{j,u}(\ell)}{\sum_{q=1}^r R_{q,u}(\ell)}. \quad (24)$$

In this case, the graph at position u loses j edges of right degree j and gains $j-1$ edges of right degree $j-1$, and the expected graph evolution at position $u \neq m$ is

$$\begin{aligned} \mathbb{E}[R_{j,u}(\ell+1) - R_{j,u}(\ell) | \text{pos}(\ell) = m] \\ = j \xi_{m,u}(\ell) \frac{R_{j+1,u}(\ell) - R_{j,u}(\ell)}{\sum_{q=1}^r R_{q,u}(\ell)}, \end{aligned} \quad (25)$$

where for $j = r$, $R_{j+1,u}(\ell) = 0$. In addition, we lose a variable node at position u with probability $\lambda_{m,u}(\ell)$. Therefore

$\mathbb{E}[V_u(\ell+1) - V_u(\ell) | \text{pos}(\ell) = m] = -\lambda_{m,u}(\ell)$. Finally, for $2 \leq j \leq r$, the expected graph evolution after one iteration of PD is given by

$$\begin{aligned} \mathbb{E}[R_{j,u}(\ell+1) - R_{j,u}(\ell)] \\ = j \sum_{\substack{m=1 \\ m \neq u}}^D \xi_{m,u}(\ell) \frac{R_{j+1,u}(\ell) - R_{j,u}(\ell)}{\sum_{q=1}^r R_{q,u}(\ell)} p_m(\ell) \\ = j \left(\frac{R_{j+1,u}(\ell) - R_{j,u}(\ell)}{\sum_{q=1}^r R_{q,u}(\ell)} \right) (p \xi_u^T - p_u(\ell)), \end{aligned} \quad (26)$$

where $p \doteq [p_1(\ell) \dots p_D(\ell)]$ and $\xi_u \doteq [\xi_{1,u}, \xi_{2,u}, \dots, \xi_{D,u}]$. For the case $j = 1$, we only have to include the degree-one check node removed if position u is selected. Thus

$$\begin{aligned} \mathbb{E}[R_{1,u}(\ell+1) - R_{1,u}(\ell)] = \\ = -p_u(\ell) + \left(\frac{R_{2,u}(\ell) - R_{1,u}(\ell)}{\sum_{q=1}^r R_{q,u}(\ell)} \right) (p \xi_u^T - p_u(\ell)), \end{aligned} \quad (27)$$

and for the variable nodes we obtain $w \mathbb{E}[V_u(\ell+1) - V_u(\ell)] = -p \lambda_u^T$, where $\lambda_u \doteq [\lambda_{1,u}, \lambda_{2,u}, \dots, \lambda_{D,u}]$.